

Buletin Ilmiah Mat. Stat. dan Terapannya (Bimaster)
Volume 08, No. 4 (2019), hal 705-714.

PENERAPAN ALGORITMA *BRANCH AND BOUND* DALAM MENENTUKAN OPTIMASI JUMLAH PRODUKSI ROTI (Studi Kasus: CV Sedap Sari Bakery)

Meliana, Evi Noviani, Hendra Perdana

INTISARI

Perusahaan Sedap Sari Bakery merupakan salah satu produsen makanan yang berada di Kota Singkawang. Setiap perusahaan yang bergerak dalam bidang industri, pasti menginginkan hasil produksi yang maksimal dengan memperhatikan keterbatasan sumber daya yang ada. Masalah kombinasi produk yang sering dihadapi perusahaan, yaitu menentukan jumlah masing-masing produk yang harus diproduksi. Tujuan dari penelitian ini adalah menganalisis kombinasi jumlah produksi roti di perusahaan Sedap Sari Bakery sehingga mendapatkan hasil yang maksimal menggunakan algoritma branch and bound. Kasus ini dapat dimodelkan dalam bentuk program linier untuk mengoptimalkan produksi roti berupa bilangan bulat. Dalam menyelesaikan masalah program linier tersebut maka metode yang digunakan adalah metode simpleks. Hasil penghitungan yang diperoleh menggunakan metode simpleks tidak selalu bernilai bilangan bulat. Oleh karena itu, untuk mendapatkan solusi bilangan bulat maka digunakan algoritma branch and bound. Perusahaan Sedap Sari Bakery memproduksi enam jenis roti yaitu roti bolu, roti gulung, roti mandarin, roti tiga rasa, roti coklat dan roti isi coklat. Data yang digunakan dalam penelitian ini berupa data persediaan bahan baku roti dalam waktu satu hari dari keenam jenis roti tersebut. Dari penghitungan algoritma branch and bound diperoleh jumlah kombinasi produk roti yaitu roti bolu sebanyak 423 bungkus, roti gulung sebanyak 166 bungkus, roti mandarin sebanyak 100 bungkus, roti tiga rasa sebanyak 175 bungkus, roti coklat sebanyak 423 bungkus dan roti isi coklat 342 bungkus, dengan keuntungan maksimal sebesar Rp5.101.100.

Kata kunci: *metode simpleks, algoritma branch and bound, program linier*

PENDAHULUAN

Produksi barang atau jasa merupakan salah satu bidang usaha yang sedang berkembang pada saat ini. Salah satu perusahaan yang sedang berkembang adalah perusahaan Sedap Sari Bakery. Perusahaan Sedap Sari Bakery merupakan salah satu produsen makanan yang berada di Kota Singkawang. Perusahaan Sedap Sari Bakery adalah perusahaan yang berdiri sejak tahun 2005 dan bergerak dalam bidang pembuatan roti.

Pada dasarnya tujuan dari setiap perusahaan adalah untuk memperoleh keuntungan yang maksimal. Masalah kombinasi produk yang sering dihadapi perusahaan, yaitu menentukan jumlah masing-masing produk yang harus diproduksi. Hal yang perlu dipertimbangkan dalam penyusunan perencanaan produksi yaitu mengoptimalkan produksi sehingga akan dapat dicapai tingkat biaya yang paling minimum untuk pelaksanaan proses produksi. Oleh karena itu, perusahaan membutuhkan solusi untuk mengoptimalkan produksi dengan memperhatikan keterbatasan-keterbatasan sumber daya yang ada. Untuk penyelesaian masalah optimasi digunakan *linear programming*.

Optimasi produksi yang baik harus diketahui besarnya permintaan pasar, sehingga memudahkan perusahaan mengetahui jumlah produk yang harus diproduksi. Program linier merupakan suatu model matematis untuk mendapatkan alternatif penggunaan terbaik atas sumber-sumber yang tersedia [1]. Untuk mendapatkan solusi yang optimal, maka algoritma yang digunakan adalah algoritma *branch and bound*.

Algoritma *branch and bound* merupakan suatu algoritma untuk menghasilkan penyelesaian optimal pemrograman linier yang menghasilkan variabel-variabel keputusan yang berupa bilangan bulat. Algoritma *branch and bound* membatasi penyelesaian optimal yang akan menghasilkan bilangan pecahan dengan cara membuat cabang atas dan bawah bagi masing-masing variabel keputusan yang bernilai pecahan agar bernilai bulat sehingga setiap pembatas akan menghasilkan cabang baru [2].

Penelitian ini bertujuan untuk menganalisis dan menentukan nilai kombinasi jumlah produksi roti di perusahaan Sedap Sari Bakery sehingga mendapatkan hasil yang maksimal menggunakan algoritma *branch and bound*. Perhitungan optimasi pada penelitian ini difokuskan pada hasil yang berupa bilangan bulat. Hal-hal yang berhubungan dengan pengadaan bahan baku dianggap selalu ada, serta distribusi roti ke toko-toko diabaikan. Produk roti yang dianalisis sebanyak 6 jenis roti yaitu roti bolu, roti gulung, roti mandarin, roti tiga rasa, roti coklat dan roti isi coklat.

Langkah pertama dalam penelitian ini adalah menyelesaikan masalah pemrograman linier dengan metode simpleks. Selanjutnya mencari dan menentukan nilai optimal dengan melihat hasil yang diperoleh sudah bernilai bilangan bulat atau belum. Langkah selanjutnya adalah nilai solusi yang bernilai tidak bulat yang layak dicabangkan ke dalam sub-sub masalah. Langkah terakhir adalah melakukan proses percabangan sampai diperoleh hasil yang bernilai bilangan bulat dari setiap sub-sub masalah.

Pemrograman Linier

Pemrograman linier merupakan sebuah metode matematis yang berbentuk linier untuk menemukan suatu penyelesaian optimal dengan cara memaksimumkan atau meminimumkan fungsi tujuan terhadap suatu susunan kendala [2]. Fungsi tujuan adalah fungsi yang akan dicari nilai optimumnya yang dinyatakan sebagai Z . Sedangkan fungsi kendala merupakan bentuk matematis Batasan kapasitas yang tersedia. Bentuk umum dari model matematis pemrograman linier adalah sebagai berikut:

Fungsi tujuan:

$$\text{Memaksimumkan atau meminimumkan: } Z = \sum_{j=1}^n c_j x_j \quad (1)$$

Fungsi kendala:

$$\sum_{j=1}^n a_{ij} x_j (\leq, =, \geq) b_i, x_j \geq 0$$

dengan,

n : Banyaknya variabel keputusan mulai dari 1, 2, ..., n

Z : Fungsi tujuan yang akan di cari nilai optimalnya (maksimum atau minimum)

c_j : Parameter fungsi tujuan ke- j , untuk $j = 1, 2, \dots, n$.

x_j : Banyaknya kegiatan ke- j , untuk $j = 1, 2, \dots, n$.

a_{ij} : Banyaknya sumber daya ke- i yang diperlukan untuk menghasilkan setiap unit keluaran Kegiatan, untuk $j(i = 1, 2, \dots, m)$ dan $j = 1, 2, \dots, n$.

b_i : Sumber yang tersedia dalam kendala i , untuk $i = 1, 2, \dots, m$

Metode yang digunakan untuk mencari solusi dari model pemrograman linier adalah metode simpleks. Berikut ini akan dijelaskan mengenai metode simpleks.

Metode Simpleks

Metode simpleks merupakan suatu proses dalam menyelesaikan masalah pemrograman linier untuk mencari penyelesaian optimal melalui perhitungan ulang (iterasi) yang disesuaikan dengan fungsi tujuan dan sejumlah kendala tertentu. Dalam menentukan apakah variabel keputusan telah menghasilkan penyelesaian optimal, maka metode simpleks dapat disajikan ke dalam sebuah tabel simpleks. Berikut adalah tabel untuk perhitungan metode simpleks [3].

Tabel 1 Tabel Simpleks

C_{Bi}	Vektor dalam Basis	C_j	C_1	C_2	...	C_j	...	C_n	R_i
		b_i	b_1	b_2	...	b_j	...	b_n	
C_{B1}	S_1	b_1	a_{11}	a_{21}	...	a_{j1}	...	a_{n1}	R_1
C_{B2}	S_2	b_2	a_{12}	a_{22}	...	a_{j2}	...	a_{n2}	R_2
.
.
.
C_{Bn}	S_n	b_n	a_{1n}	a_{2n}	...	a_{jn}	...	a_{nn}	R_n
	Z	0	Z	Z	...	Z	...	Z	
	$Z - C_j$	0	$Z - C_1$	$Z - C_2$...	$Z - C_j$...	$Z - C_n$	

Keterangan:

1. Kolom pertama dari tabel yaitu C_{Bi} menggambarkan koefisien biaya relatif untuk variabel dalam basis.
2. Kolom kedua memberikan vektor basis berisikan variabel *slack*, variabel ini akan digantikan dengan variabel keputusan.
3. Kolom ketiga C_j berisikan konstanta ruas kanan setiap batasan atau konstanta fungsi kendala, untuk $j = 1, 2, \dots, n$.
4. Baris C_j berisikan koefisien biaya relatif dari fungsi tujuan, dan untuk kolom variabel *slack* bernilai nol.
5. Kolom kunci merupakan kolom yang akan dicari dengan melihat nilai negatif terkecil yang ada pada baris $Z - C_j$.
6. Kolom R_i yaitu persamaan $R_i = \frac{C_j b_i}{\text{Kolom kunci}}$, untuk mencari nilai indeks.
7. Baris a_j berisikan variabel keputusan dan variabel *slack*.
8. Baris Z diisi dengan rumus $Z = \sum_{i=1}^n b_i \cdot a_{ji}$, untuk $j = 1, 2, \dots, n$.
9. Baris $Z - C_j$ berisikan angka hasil pengurangan antara $Z - C_j$. Baris ini yang akan memberikan informasi apakah fungsi tujuan telah optimal atau belum. Tabel dikatakan telah optimal jika semua nilai pada baris $Z - C_j \geq 0$ untuk tujuan memaksimumkan, sedangkan untuk tujuan meminimumkan nilai pada baris $Z - C_j \leq 0$.

Kendala yang terdapat dalam fungsi kendala model pemrograman linier dapat dibedakan dengan tanda hubungan matematis berupa \leq (pertidaksamaan kurang dari), $=$ (persamaan sama dengan), dan \geq (pertidaksamaan lebih dari). Berdasarkan hal tersebut maka dalam metode simpleks perlu ditambahkan ruas kiri setiap kendala dalam fungsi kendala model program linier variabel yang dikenal sebagai variabel penolong, yang terdiri dari variabel *slack*, variabel *surplus*, dan variabel *artificial* [4].

Variabel *slack* digunakan untuk mengkonversikan pertidaksamaan (\leq) sampai menjadi sama dengan ($=$) dengan menambahkan variabel *slack* kesisi kiri batasan. Variabel *slack* akan berfungsi sebagai variabel basis pada solusi awal. Pertidaksamaan kendala pembatas:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \text{ untuk } i = 1, 2, \dots, m$$

diubah menjadi,

$$\sum_{j=1}^n a_{ij} x_j + S_i = b_i \quad (2)$$

dengan:

x_j : variabel keputusan ke- j , untuk $j = 1, 2, \dots, n$

a_{ij} : koefisien fungsi kendala ke- i variabel keputusan ke- j

S_i : *Slack* variabel kendala ke- i , untuk $i = 1, 2, \dots, n$

b_i : nilai ruas kanan kendala ke- i , untuk $i = 1, 2, \dots, n$

Variabel *surplus* digunakan untuk mengkonversikan pertidaksamaan (\geq) lebih besar sama dengan menjadi persamaan sama dengan ($=$) dengan mengurangi variabel *surplus* kesisi kiri batasan. Pertidaksamaan kendala:

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \text{ untuk } i = 1, 2, \dots, m$$

diubah menjadi

$$\sum_{j=1}^n a_{ij} x_j - S_i + d_i = b_i \quad (3)$$

dengan:

x_j : variabel keputusan ke- j , untuk $j = 1, 2, \dots, n$

a_{ij} : koefisien fungsi kendala ke- i variabel keputusan ke- j

S_i : *surplus* variabel kendala ke- i , untuk $i = 1, 2, \dots, n$

b_i : nilai ruas kanan kendala ke- i , untuk $i = 1, 2, \dots, n$

d_i : *artificial* variabel ke- i , untuk $i = 1, 2, \dots, n$

Artificial variabel adalah variabel yang difungsikan sebagai variabel basis awal untuk ditambahkan ke model matematik kendala dengan bentuk \geq atau $=$. Gagasan penggunaan *artificial* variabel sangat sederhana, dengan menambahkan suatu *artificial* variabel pada sisi kiri setiap persamaan [5]. Pada solusi optimum variabel *artificial* harus bernilai nol, jika tidak bernilai nol, maka solusi yang dihasilkan akan menjadi tidak layak.

Persamaan kendala:

$$\sum_{j=1}^n a_{ij} x_j = b_i, \text{ untuk } i = 1, 2, \dots, m$$

diubah menjadi

$$\sum_{j=1}^n a_{ij} x_j + d_i = b_i \quad (4)$$

dengan:

x_j : variabel keputusan ke- j , untuk $j = 1, 2, \dots, n$

a_{ij} : koefisien fungsi kendala ke- i variabel keputusan ke- j

b_i : nilai ruas kanan kendala ke- i , untuk $i = 1, 2, \dots, n$

d_i : *artificial* variabel ke- i , untuk $i = 1, 2, \dots, n$

Pemrograman Linier *Integer*

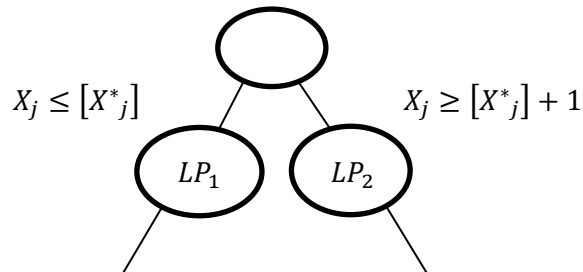
Pemrograman linier *integer* merupakan suatu proses dari pemecahan pemrograman linier yang bertujuan untuk mencari solusi dengan nilai bulat (*integer*). Pemrograman linier *integer* digunakan untuk memodelkan permasalahan yang variabel-variabelnya harus berupa bilangan bulat, seperti variabel yang mempresentasikan jumlah orang, karena jumlah orang pasti bulat dan tidak mungkin berupa pecahan [6]. Untuk dapat memperoleh hasil yang optimal dan berupa bilangan bulat maka algoritma yang digunakan adalah algoritma *branch and bound*.

Algoritma *Branch and Bound*

Algoritma *branch and bound* pertama kali diperkenalkan oleh Land dan Doig pada tahun 1960. Algoritma *branch and bound* merupakan suatu metode untuk menghasilkan penyelesaian optimal pemrograman linier yang menghasilkan variabel-variabel keputusan bilangan bulat. Misalkan X_j adalah variabel *branching* dan X_j^* sebagai nilai dalam solusi. Nilai dalam solusi tersebut dinotasikan dengan $[X_j^*]$ yang merupakan bilangan bulat terbesar yang kurang dari sama dengan X_j^* , lakukan

branching dari simpul untuk sub masalah, untuk menghasilkan dua sub-sub masalah baru dengan menambahkan

kendala X_j lebih besar sama dengan $\lceil X_j^* \rceil$ ditambah 1 [7]. Berikut ini adalah pohon solusi untuk proses pencabangan algoritma *branch and bound* untuk menyelesaikan masalah *integer programming*:



Gambar 1 Proses Percabangan Algoritma *Branch and Bound*

Langkah-Langkah Algoritma *Branch and Bound*

Adapun langkah-langkah metode *Branch and Bound* untuk masalah maksimalisasi dapat diringkas sebagai berikut:

1. Selesaikan masalah pemrograman linier dengan metode simpleks.
2. Teliti terhadap solusi optimalnya. Jika variabel basis yang diharapkan integer adalah integer, maka solusi optimal sudah tercapai. Jika salah satu atau lebih dari variabel basis yang diharapkan berbentuk integer ternyata tidak berbentuk integer, maka lanjutkan kelangkah 3.
3. Nilai solusi yang bernilai tidak bulat layak dicabangkan kedalam sub-sub masalah. Dengan tujuan untuk menghilangkan solusi yang tidak memenuhi persyaratan bulat dari masalah tersebut.
4. Untuk setiap sub-sub masalah, nilai solusi optimal fungsi tujuan dijadikan sebagai batas atas dan solusi bulat terbaik menjadi batas bawah. Pada analisis selanjutnya sub-sub masalah yang memiliki batas atas kurang dari batas bawah yang ada tidak diikuti sertakan. Suatu solusi bulat yang layak adalah solusi yang lebih baik dari batas atas untuk setiap sub-sub masalah yang dicari. Jika solusi yang demikian terjadi, maka suatu sub-sub masalah dengan batas atas terbaik dipilih untuk dicabangkan. Kemudian kembali ke langkah 3.

Penyelesaian Masalah Produksi Roti di CV Sedap Sari Bakery

Sumber data yang digunakan dalam penelitian ini merupakan data primer yang diperoleh dari Sedap Sari Bakery. Data yang diperlukan dalam penelitian ini adalah data persediaan bahan baku roti dalam waktu satu hari dengan 6 jenis roti. Kapasitas bahan maksimum yang digunakan oleh perusahaan dalam pembuatan roti yaitu sebanyak 350 kg telur, 300 kg gula pasir, 80 kg ovalett, 350 kg tepung terigu, 150 kg mentega, 75 kg meses, 50 kg susu, dan 150 kg coklat pasta. Adapun bahan yang digunakan dalam pembuatan masing-masing jenis roti dapat dilihat pada Tabel 2 yaitu sebagai berikut:

Tabel 2 Data Penelitian

Bahan yang dibutuhkan	Bolu (Kg)	Gulung (Kg)	Mandarin (Kg)	Tiga rasa (Kg)	Coklat (Kg)	Isi Coklat (Kg)	Persediaan (Kg)
Telur	0,2500	0,2500	0,4000	0,2500	0,2000	0,1000	350
Gula Pasir	0,1875	0,3000	0,3000	0,1875	0,1500	0,1300	300
Ovalett	0,0400	0,0500	0,0500	0,0400	0	0,1250	80
Terigu	0,1250	0,2000	0,1200	0,1250	0,0750	0,5000	350
Mentega	0,0625	0,1000	0,1300	0,0625	0,1250	0,0600	150
Meses	0,1000	0	0,1200	0,0380	0	0	75

L	Susu	0,0500	0,0500	0,0500	0,0300	0	0,0300	50
ang kah	Coklat Pasta	0	0	0,1000	0	0,2500	0,1000	150

awal dalam memodelkan suatu masalah yaitu dengan menentukan variabel-variabel yang berpengaruh di dalamnya. Misalkan variabel keputusan dinotasikan sebagai x_1, x_2, x_3, x_4, x_5 dan x_6 . dengan:

- x_1 : Banyaknya roti bolu yang diproduksi setiap hari,
- x_2 : Banyaknya roti gulung yang diproduksi setiap hari,
- x_3 : Banyaknya roti mandarin yang diproduksi setiap hari,
- x_4 : Banyaknya roti tiga rasa yang diproduksi setiap hari,
- x_5 : Banyaknya roti coklat yang diproduksi setiap hari,
- x_6 : Banyaknya roti isi coklat yang diproduksi setiap hari.

Tujuan dari masalah kombinasi produk adalah untuk memaksimalkan keuntungan yang diperoleh perusahaan Sedap Sari Bakery. Penetapan harga jual oleh perusahaan dengan melihat harga pasar sedangkan biaya produksi diperoleh dari harga pokok produksi. Harga jual yang ditetapkan oleh perusahaan pada setiap jenis roti dapat dilihat pada Tabel 3 berikut:

Tabel 3 Harga Jual Tiap Jenis Roti

No	Jenis roti	Harga Jual Roti
1	Bolu	Rp3.300
2	Gulung	Rp3.600
3	Mandari	Rp2.200
4	Tiga rasa	Rp2.800
5	Coklat	Rp3.000
6	Isi coklat	Rp3.300

Perusahaan menetapkan harga jual setiap jenis roti dengan harga Rp3.300 untuk roti bolu, roti gulung seharga Rp3.600, roti Mandarin seharga Rp2.200, roti tiga rasa seharga Rp2.800, roti coklat seharga Rp3.000 dan roti isi coklat seharga Rp3.300. Selain batasan pada bahan, target produksi yang di tetapkan perusahaan juga diperhatikan. Batasan minimal yang diproduksi oleh perusahaan adalah sebanyak 100 pcs untuk roti bolu, 100 pcs untuk roti gulung, 100 pcs untuk roti mandarin, 100 pcs untuk roti tiga rasa, 100 pcs untuk roti coklat dan 100 pcs untuk roti isi coklat. Data bahan baku dan banyaknya bahan baku yang disediakan oleh perusahaan untuk memproduksi setiap jenis roti diformulasikan ke dalam model matematis.

Bentuk model matematis pemrograman linier dari permasalahan optimasi jumlah produksi roti untuk memperoleh hasil yang maksimal adalah sebagai berikut:

fungsi tujuan:

$$\text{maksimumkan } Z = 3.300 x_1 + 3.600 x_2 + 2.200 x_3 + 2.800 x_4 + 3.000 x_5 + 3.300 x_6$$

fungsi kendala:

$$0,25x_1 + 0,25x_2 + 0,4x_3 + 0,25x_4 + 0,2x_5 + 0,1x_6 \leq 350$$

$$0,1875x_1 + 0,3x_2 + 0,3x_3 + 0,1875x_4 + 0,15x_5 + 0,13x_6 \leq 300$$

$$0,04x_1 + 0,05x_2 + 0,05x_3 + 0,04x_4 + 0,125x_6 \leq 80$$

$$0,125x_1 + 0,2x_2 + 0,12x_3 + 0,125x_4 + 0,075x_5 + 0,5x_6 \leq 350$$

$$0,0625x_1 + 0,1x_2 + 0,13x_3 + 0,0625x_4 + 0,125x_5 + 0,06x_6 \leq 150$$

$$0,1x_1 + 0,12x_3 + 0,038x_4 \leq 75$$

$$0,05x_1 + 0,05x_2 + 0,05x_3 + 0,03x_4 + 0,03x_6 \leq 50$$

$$0,1x_3 + 0,25x_5 + 0,1x_6 \leq 150$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 100$$

Bentuk pemrograman linier tersebut disusun dalam tabel awal simpleks. Berikut ini adalah tabel simpleks data perusahaan Sedap Sari Bakery yakni sebagai berikut:

Tabel 4 Tabel awal simpleks

C_{Bi}	VdB	C_j	3.300	3.600	2.200	2.800	3.000	3.300	0	0	0	0	0	0	0	0	Ratio
		b_i	x_1	x_2	x_3	x_4	x_5	x_6	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	
0	S_1	350	0,2500	0,2500	0,4000	0,2500	0,2000	0,1000	1	0	0	0	0	0	0	0	1400
0	S_2	300	0,1875	0,3000	0,3000	0,1875	0,1500	0,1300	0	1	0	0	0	0	0	0	1000
0	S_3	80	0,0400	0,0500	0,0500	0,0400	0	0,1250	0	0	1	0	0	0	0	0	1600
0	S_4	350	0,1250	0,2000	0,1200	0,1250	0,0750	0,5000	0	0	0	1	0	0	0	0	1750
0	S_5	150	0,0625	0,1000	0,1300	0,0625	0,1250	0,0600	0	0	0	0	1	0	0	0	1500
0	S_6	75	0,1000	0	0,1200	0,0380	0	0	0	0	0	0	0	1	0	0	0
0	S_7	50	0,0500	0,0500	0,0500	0,0300	0	0,0300	0	0	0	0	0	0	1	0	1000
0	S_8	150	0	0	0,1000	0	0,2500	0,1000	0	0	0	0	0	0	0	1	0
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	$Z - C_j$	0	-3.300	-3.600	-2.200	-2.800	-3.000	-3.300									

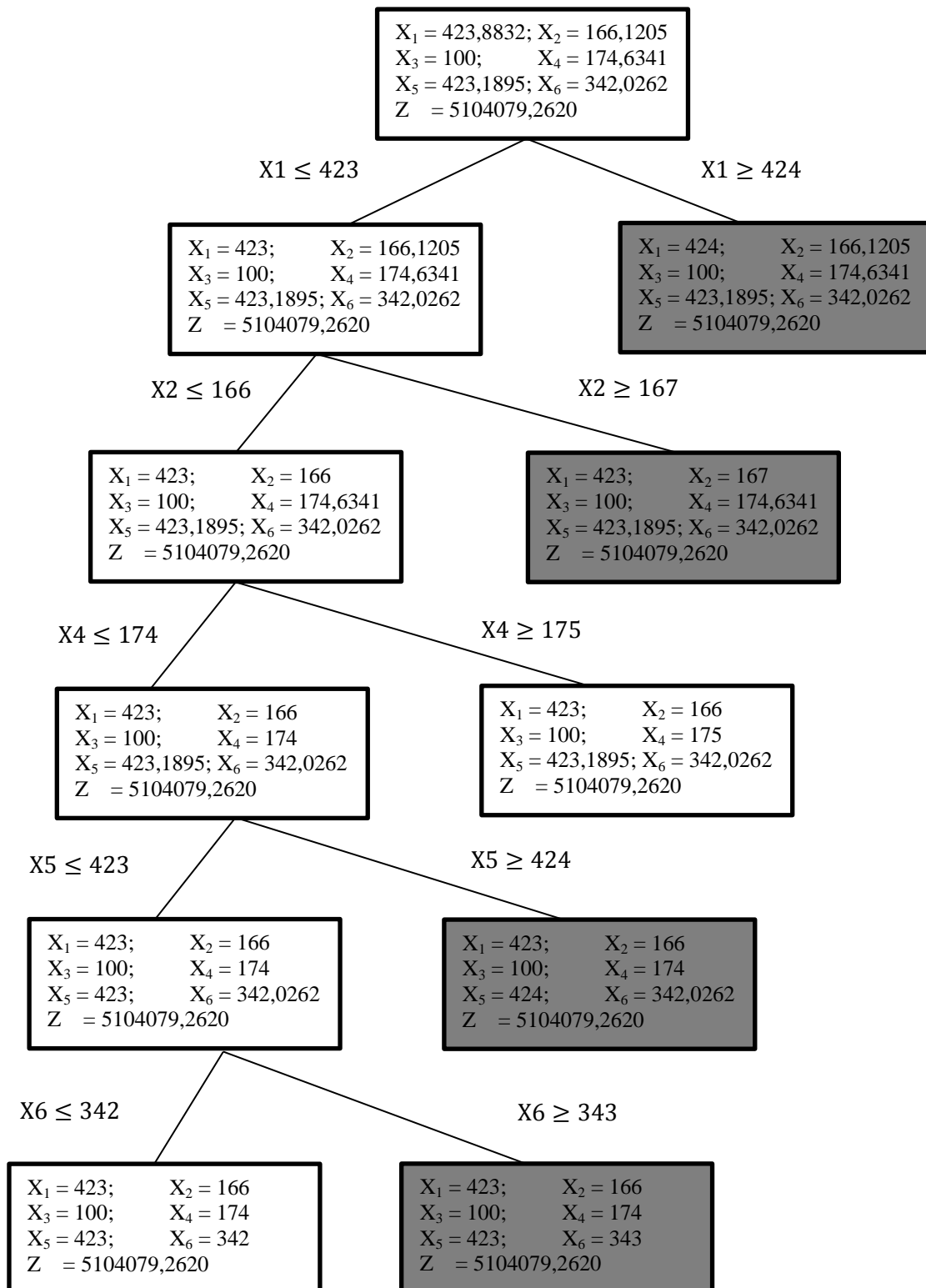
Dari tabel tersebut akan dilihat nilai pada baris $Z - C_j$ apakah sudah optimal atau belum. tabel dikatakan telah optimal jika semua nilai pada $Z - C_j$ sudah bernilai positif. Pada penelitian ini nilai $Z - C_j$ tabel simpleks masih terdapat nilai negatif, maka tabel dikatakan belum optimal. Dilanjutkan langkah-langkah perbaikan atau perubahan pada tabel simpleks yang belum optimal sampai sampai diperoleh hasil yang optimal.

Penyelesaian permasalahan optimasi jumlah produksi untuk memperoleh hasil yang maksimal pada contoh kasus Sedap Sari Bakery dengan menggunakan metode simpleks adalah sebagai berikut: $x_1 = 423,8833$, $x_2 = 166,1205$, $x_3 = 100$, $x_4 = 174,6341$, $x_5 = 423,1895$, $x_6 = 342,0262$ dengan $Z = 5104079,2620$. Hasil yang diperoleh dalam perhitungan tersebut masih terdapat nilai yang berupa bilangan tidak bulat.

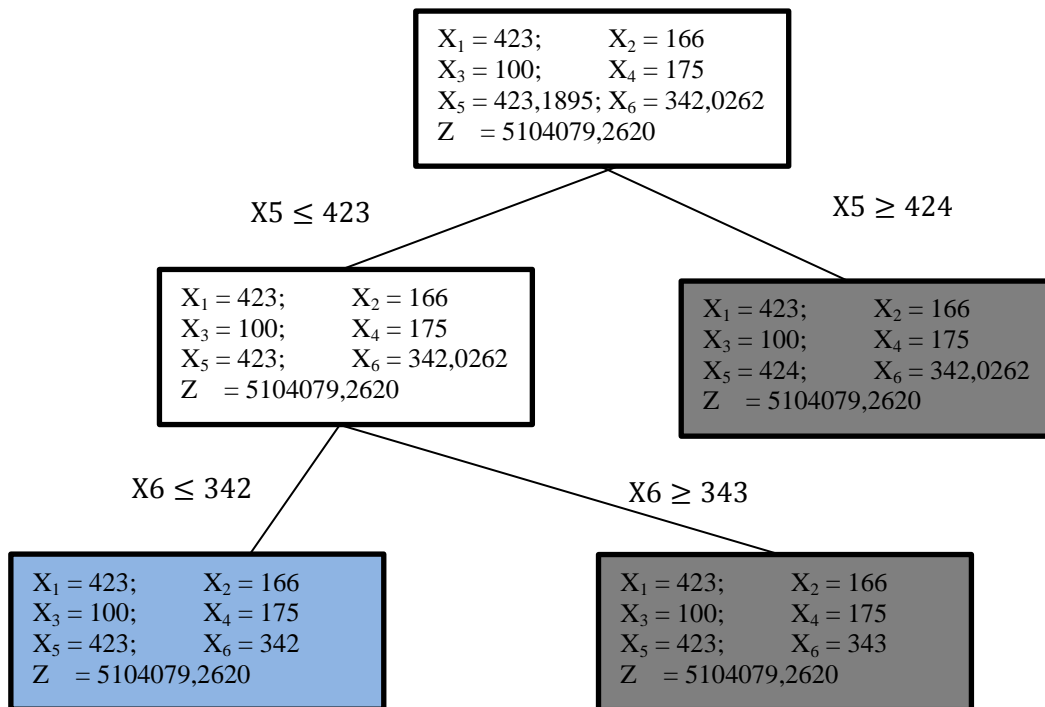
Karena solusi optimal yang diperoleh bukan merupakan bilangan bulat, maka nilai solusi yang tidak bulat dicabangkan ke dalam sub-sub masalah. Algoritma *branch and bound* merupakan salah satu algoritma yang dapat menyelesaikan pemrograman linier yang menghasilkan variabel-variabel bilangan bulat. Pada solusi optimal dari metode simpleks variabel pertama yang memiliki bilang tidak bulat adalah variabel x_1 , sehingga x_1 menjadi variabel percabangan (*branching*) untuk iterasi 1. Dari variabel percabangan ini kemudian akan menghasilkan dua sub masalah yaitu untuk $x_1 \leq 423$ dan $x_1 \geq 424$. Kemudian dari kedua sub masalah tersebut hasil yang layak untuk dicabangkan selanjutnya adalah untuk $x_1 \leq 423$ karena merupakan solusi yang layak.

Variabel $x_1 \leq 423$ akan dilanjutkan untuk percabangan iterasi kedua dengan variabel yang akan dicabangkan adalah variabel x_2 , untuk $x_2 \leq 166$ dan $x_2 \geq 167$. Hasil iterasi tersebut menunjukkan bahwa untuk $x_2 \leq 166$ merupakan solusi layak yang akan digunakan untuk iterasi ketiga. variabel $x_2 \leq 166$ dilanjutkan percabangan untuk variabel x_4 dengan $x_4 \leq 174$ dan $x_4 \geq 175$. Hasil dari iterasi ketiga menunjukkan bahwa solusi yang diperoleh untuk $x_4 \leq 174$ dan $x_4 \geq 175$ merupakan solusi yang layak. Kemudian $x_4 \leq 174$ dan $x_4 \geq 175$ digunakan untuk melanjutkan percabangan iterasi keempat. Untuk iterasi keempat percabangan selanjutnya adalah percabangan untuk $x_5 \leq 423$ dan $x_5 \geq 424$. Dari iterasi keempat tersebut yang merupakan solusi layak untuk dicabangkan adalah $x_5 \leq 423$. Selanjutnya untuk iterasi kelima yang akan dicabangkan adalah variabel x_6 untuk $x_6 \leq 342$ dan $x_6 \geq 343$. Dari hasil iterasi kelima untuk $x_6 \leq 342$ merupakan solusi optimal yang diperoleh

algoritma *branch and bound*. Hasil ini dirangkum dalam pohon solusi iterasi pertama sampai iterasi kelima. Pohon solusi algoritma *branch and bound* adalah sebagai berikut:



Gambar 2 Pohon Solusi Iterasi Pertama, Kedua, Ketiga, Keempat dan Kelima dengan Algoritma *Branch and Bound*



Gambar 4 Pohon Solusi Iterasi Ketiga, Keempat dan Kelima untuk $X_4 \geq 175$ dengan Algoritma *Branch and Bound*

Warna abu-abu pada pohon solusi di atas menunjukkan hasil yang tidak layak. Hal ini dikarenakan hasil perhitungan terlihat bahwa penggunaan bahan yang digunakan melebihi persediaan bahan yang telah ditentukan. Sedangkan yang berwarna putih merupakan hasil yang layak, karena hasil perhitungan terlihat bahwa bahan maksimal yang digunakan tidak melebihi persediaan yang ada. Hasil yang berwarna biru merupakan hasil optimal yang diperoleh dari algoritma *branch and bound*.

Penyelesaian permasalahan optimasi jumlah produksi roti dengan menggunakan algoritma *branch and bound* menghasilkan penyelesaian yang bernilai bulat dengan $x_1 = 423$, $x_2 = 167$, $x_3 = 100$, $x_4 = 174$, $x_5 = 423$, $x_6 = 342$ dengan $Z = 5101100$. Solusi tersebut menunjukkan bahwa Perusahaan Sedap Sari Bakery dapat menghasilkan jumlah kombinasi roti yaitu roti bolu sebanyak 423 bungkus, roti gulung sebanyak 166 bungkus, roti mandarin sebanyak 100 bungkus, roti tiga rasa sebanyak 175 bungkus, roti coklat sebanyak 423 bungkus dan roti isi coklat 342 bungkus dengan keuntungan maksimal yang diperoleh yaitu sebesar Rp5.101.100.

KESIMPULAN

Berdasarkan hasil pembahasan pada bab sebelumnya, maka dapat disimpulkan bahwa penerapan algoritma *branch and bound* dalam mencari keuntungan optimal produksi roti perhari menghasilkan keuntungan sebesar Rp5.101.100. Jumlah kombinasi produk roti yaitu roti bolu sebanyak 423 pcs, roti gulung sebanyak 166 bungkus, roti mandarin sebanyak 100 bungkus, roti tiga rasa sebanyak 175 bungkus, roti coklat sebanyak 423 bungkus dan roti isi coklat 342 bungkus. Hasil perhitungan menggunakan algoritma *branch and bound* merupakan solusi yang bernilai bilangan bulat. Sesuai dengan kebutuhan perusahaan dalam menentukan jumlah produk roti yang harus diproduksi agar

mencapai hasil yang maksimal dan berupa bilangan bulat. Namun prosedur dari algoritma ini sangat panjang karena algoritma ini mengharuskan pemecahan program linier yang lengkap pada setiap proses. Dalam masalah dengan variabel yang lebih banyak, algoritma ini akan menghasilkan percabangan lebih banyak sehingga proses perhitungan akan memakan banyak waktu.

DAFTAR PUSTAKA

- [1] Aminudin, A. *Prinsip-Prinsip Riset Operasi*. Jakarta: Erlangga. 2005.
- [2] Siswanto. *Operations Research*. Jilid 1. Bogor: Erlangga. 2006.
- [3] Supranto, J. *Linear Programming*. Edisi Kedua. Fakultas Ekonomi Universitas Indonesia. Jakarta. 1983.
- [4] Sitinjak, T.J.R. *Riset Operasi untuk Pengambilan Keputusan Manajerial dengan Aplikasi Excel*. Graha Ilmu. Yogyakarta. 2006.
- [5] Mulyono, S. *Riset Operasi*. Edisi Revisi, Fakultas Ekonomi Universitas Indonesia. Jakarta. 2004.
- [6] Aprilia, S. Aplikasi Algoritma *Branch and Bound* untuk Menyelesaikan *Integer Programming*. *Jurnal informatika*. 2009.
- [7] Hilier, F.S. dan Lieberman, G.J.. *Introduction To Operations Research*. Edisi Pertama. Andi. Yogyakarta. 2008.

MELIANA : Jurusan Matematika FMIPA UNTAN, Pontianak
melianadk10@gmail.com

EVI NOVIANI : Jurusan Matematika FMIPA UNTAN, Pontianak
evi_noviani@math.untan.ac.id

HENDRA PERDANA : Jurusan Matematika FMIPA UNTAN, Pontianak
hendra.perdana@math.untan.ac.id
